Computational Economics (2025) 65:3923–3943
https://doi.org/10.1007/s10614-024-10697-z

Detecting Insider Trading in the Indian Stock Market: An Optimized Deep Learning Approach

Prashant Priyadarshi[1] · **Prabhat Kumar**[1]

Accepted: 1 August 2024 / Published online: 9 August 2024

Abstract

A novel approach is proposed in this study for identifying insider trading in the Indian stock market by classifying multiple multivariate time series financial data using deep learning. The model utilizes multi-channel convolutional neural network (MTC-CNN) and MTC-CNN with Optuna hyperparameter optimization. In order to test the method, insider trading samples from 2001 to 2020 are used, along with corresponding non-insider trading samples from the same period. As a result of our experiments, we found that under the following conditions of 30-, 60-, and 90-day time window lengths, the accuracy of the proposed method are 87.50%, 75.00%, and 62.50%, respectively. It has also been found that using OPTUNA hyperparameter optimization, the false positive rate was reduced by 20% for all the time windows. These accuracy rates surpass those of the benchmark models like logistic regression, random forest, and convolutional neural network, providing evidence that the proposed system is effective in identifying the activities of insider traders. The proposed deep learning model serves as a valuable tool for market regulators and investors in detecting and preventing illicit trading practices, ultimately fostering integrity and fairness in the Indian securities market.

Keywords Indian stock market · Insider trading identification · Multi-channel CNN · OPTUNA

✉ Prashant Priyadarshi
prashantp.phd19.cs@nitp.ac.in

Prabhat Kumar
prabhat@nitp.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology, Patna, Bihar 800005, India

1 Introduction

The Indian security market is growing in importance as a component of the global financial system. Savings are transformed into investments for investors through capital markets, and business experience is transformed into funding for enterprises. The Indian securities market has seen significant growth in recent years, with increasing participation from retail investors and new financial products and services development. Like all financial markets, the Indian securities market is subject to risks such as market volatility, regulatory changes, and numerous security frauds. Several factors adversely affect the fairness of security trading, including insider trading behaviour, which occurs when individuals or organizations have obtained non-public information before its announcement and trade in a company's stock. Insider trading in the stock market yields several critical problems that collectively compromise market integrity and fairness. Firstly, it distorts the market's natural price discovery mechanism by introducing an artificial influence based on non-public information, leading to skewed stock prices that mislead other investors. This practice also grants insiders an unjust advantage, undermining the level playing field essential for fair and transparent markets. Consequently, investor confidence dwindles as the perception of a manipulated or unfair system takes root, potentially reducing market participation and efficiency. Moreover, insider trading not only violates securities laws but also breaches ethical standards, as it betrays the trust and fiduciary responsibilities insiders owe to their companies and shareholders. It further poses risks of resource misallocation, distorting investment decisions and potentially harming companies' long-term growth. Lastly, it contributes to wealth inequality, favouring in-siders' gains at the expense of investors without access to such privileged information. Regulatory efforts globally aim to curb these issues, implementing stringent measures to enforce transparency and prevent market abuse, thus fostering a more equitable and trustworthy financial landscape.

In India, the Securities and Exchange Board of India (SEBI) regulates insider trading through its Insider Trading Regulations. It is prohibited by the Companies Act, 2013 and the SEBI Act, 1992 (Securities and Exchange Board of India https://www.sebi.gov.in/). SEBI has formed the SEBI (Prohibition of Insider Trading) Regulations, 2015 which prescribe the rules of prohibition and restriction of insider trading in India. The Regulations are applicable mainly to "dealing in securities" which involves "buying, selling or agreeing to buy, sell or deal in any securities by any person either as principal or agent, by insiders on the basis of any private confidential information". Contravening any provision of the SEBI Regulations is an offense under the Act, punishable by imprisonment. Despite, the annual report of SEBI for 2021–2022 highlighted that insider trading cases constituted 34% of all investigations conducted by SEBI, and there's a noticeable upward trend in insider trading offenses each year. Even though there are strict rules and regulations against insider trading, people are still enticed by the possibility of large financial gains. Today's markets and trading practices are so complex and large that it is difficult to detect such events in a large population of lawful trades. Research has been conducted extensively to detect insider trading in an institution of national importance, but when it comes to predicting, this research has been limited to a very small number of institutions and accurate predictions remain beyond the horizon. Park and Lee (2010) developed an ARMA (1,1) model that incorporates the relationship between insider transactions and stock returns in Korean markets. Deng et al., (2019) studied the detection of insider trading in Chinese stocks based on gradient boosting decision trees and differential evolution. Deng et al., (2021) further improved their method using PCA and random forests. For the detection of insider traders in the US stock market, LSTM RNN-based deep learning coupled with discrete signal processing has been proposed as a deep learning approach on time series data (Islam, 2018). For the same market researcher, employed data mining techniques using various classical and robust outlier detection methods to detect outlying insider transactions (Esen, 2020). An alternative method of predicting insider trading events in the US market was used to predict insider trading events by using deep neural networks, consensus models, and other statistical techniques in a multistage process (Seth & Chaudhary, 2020). It is clear from the above study that modern computational technologies like artificial intelligence, machine learning, and deep learning are being adopted over statistical techniques in order to improve the methodology for real-time supervision of insider trading, which results in a more stringent and accurate securities market surveillance. Also, despite being one of the world's fastest-growing economy, little research has been conducted on insider trading in India so far. Hence, in this study, we have proposed a novel framework for the identification of insider trading in the Indian stock market.

During the last few years, deep-learning-based financial modelling has grown exponentially (Hu et al., 2021). As stock markets are multifaceted, nonlinear, and raucous, deep learning seems to be a promising approach for capturing features that are informative enough to make accurate stock market predictions. There have been a number of popular deep learning algorithms that have gained popularity in stock market prediction, including deep multilayer perceptron (MLP), restricted Boltzmann machines (RBM), long short-term memories (LSTM), autoencoders (AE), and convolutional neural networks (CNN) (Bao et al., 2017; Cai et al., 2012; Chen et al., 2015; Di Persio & Honchar, 2016; Fischer & Krauss, 2018; Gunduz et al., 2017; Yong et al., 2017). CNN always outperformed linear models (AR, MA, ARIMA, ARMA), non-linear models (ARCH, GARCH, Neural Network), and deep learning models for making predictions (Hiransha et al., 2018; Hoseinzade & Haratizadeh, 2019). However, a multi-channel CNN is more successful at handling multivariate time series data and has been applied in various fields, such as healthcare informatics and bioinformatics.

On the other hand, multichannel convolutional neural networks (CNNs) have demonstrated remarkable efficacy in handling datasets with limited sample sizes across various domains. MTC-CNN demonstrated their usefulness in small sample learning for facial expression recognition (Hamester et al., 2015; Liu et al., 2015). Successfully classification of disaster tweets and lung sound classification with lower sample sizes was made possible with MTC-CNN (Kumar et al., 2023; Messner et al., 2020). It has proven track record in industrial application, fault detection in civil infrastructure, predicting bearing remaining useful life (RUL) and surface defects on solar cells with limited data (Jiang et al., 2020; Shajihan et al., 2022; Zhang et al., 2020). These studies collectively underscore the proven capability of multichannel CNNs in effectively handling datasets of limited sizes, making them a valuable tool for various applications where data scarcity is a challenge.

As a result of CNN's proven abilities in market prediction as well as in other domains, we develop our own framework called OPTUNA hyper-parameter optimization framework for the identification of insider trading using multi-channel deep CNN. We employed OPTUNA (hyper-parameter optimization technique) to optimize the hyperparameter parameters of our classifier algorithm in order to enhance the performance of our classification system (Akiba et al., 2019). OPTUNA offers a more efficient and time-saving approach that allows one to undertake an analysis without wasting time by evaluating unnecessary parameter combinations. OPTUNA is a better alternative to random and grid search methods. In order to arrive at an optimal set of hyperparameters, it uses information from previous optimizations for intelligently exploring the parameter space in order to obtain a set of pruned hyperparameters (Arifin et al., 2021; Shekhar et al., 2021; Srinivas & Katarya, 2022). We will describe more about OPTUNA along with multi-channel CNN in the coming sections. As part of this study, samples from the Indian securities market are employed and a database of information is collected from a number of listed companies that have been convicted of insider trading by the SEBI (Security and Exchange Board of India, https://www.sebi.gov.in/) from 2001 to 2020. In the following sections, the background will be discussed in Sect. 2, the proposed method in Sect. 3. Section 4 will provide insights about the experiment design, Sect. 5 will present the experimental results, and finally, Sect. 6 will conclude the study and outline future work. The key contributions and highlights of this research are as follows:

- A novel expert system is introduced, leveraging an optimized multi-channel convolutional neural network and OPTUNA, to effectively identify insider trading.
- Among a vast array of relevant variables and indicators, 86 have been carefully selected and evaluated for their significance in identifying insider trading.
- The experiments encompass three distinct time window lengths (30, 60, and 90 days) to assess the importance of indicators in insider trading identification.
- Data preprocessing and hyperparameter tuning have been employed to enhance the performance of the expert system.
- OPTUNA, an automated expert technique, is utilized for efficient hyperparameter tuning.

2 Background

2.1 Multi-channel Deep Convolutional Neural Network (MTC-CNN)

Multi-channel CNN (MTC-CNN) is essentially a CNN with multiple input channels, where each input channel corresponds to a different type of feature or data. For example, in an image classification task, an RGB image with three colour channels can be processed by a multi-channel CNN, where each channel is treated as a separate input channel. In this case, the CNN can learn to extract different features from each colour channel, which can help to improve the overall accuracy of the classification. Similarly, in a speech recognition task, a multi-channel CNN can be used to process multiple acoustic features such as spectrograms, Mel-frequency cepstral coefficients (MFCCs), and pitch contours as separate input channels. The use of multiple input channels can provide richer information to the network, allowing it to learn more complex features and patterns in the data. This makes multi-channel CNNs a powerful tool for a wide range of applications in deep learning, including image processing, speech recognition, and natural language processing.

An MTC-CNN is composed of different layers that collectively process and extract meaningful features from the input channels. Let us explore the purpose and functionality of each layer in the network:

- Input channels: The input channels represent the different indicators or variables associated with the trading data. Each channel carries specific information that is fed into the network simultaneously.
- Convolutional layers: Convolutional layers are responsible for executing convolutions, a process where a group of filters with adjustable parameters convolve across the input channels, enabling the extraction of local patterns and features. These filters slide over the input data, applying mathematical operations to capture relevant information.
- Pooling layers: Pooling layers are utilized to decrease the spatial dimensions of the data through down sampling. They employ various techniques such as max pooling, which identifies the highest value within a given region, and average pooling, which computes the mean value. Pooling serves to simplify computational requirements and extract the most significant features from the input channels.
- Fully-connected layer: The fully-connected layer establishes connections between each neuron in the previous layer and the neurons in the subsequent layer. It combines the extracted features and acquires the ability to generate predictions using the learned representations. Ultimately, the output of this layer is usually transformed into a flattened one-dimensional vector.
- Output layer: The final predictions or classifications are generated by the output layer, which incorporates learned features. The number of neurons in this layer varies according to the task, be it binary classification or multi-class classification.

In an MTC-CNN, the convolutional and pooling layers are typically alternated multiple times before reaching the fully-connected layer. With this architecture, the network is able to learn hierarchical representations of the input data, allowing it to capture both local and global patterns, enhancing the model's ability to identify complex patterns and relationships in the context of insider trading.

In the detection of insider trading, a multi-channel CNN is used to process multiple indicators and variables associated with trading data. Each indicator or variable is treated as a separate channel, allowing the network to learn unique patterns and relationships among them. This approach helps the multi-channel CNN to capture complex interactions and dependencies among different indicators, potentially enhancing the accuracy of identifying insider trading activities. The network learns to extract important features from each channel and combines them to make predictions, enabling it to detect subtle patterns and anomalies that might indicate instances of insider trading. As illustrated in Fig. 1, the model utilizes a two-channel structure to capture both variable and indicator dimensions, offering a comprehensive perspective for enhanced financial data classification.

The use of a multi-channel CNN in detecting insider trading helps to leverage the power of deep learning and convolutional operations to effectively analyze and interpret large amounts of trading data. It allows for the detection of hidden patterns and correlations that might not be apparent through traditional statistical methods. The optimization of the network using techniques like hyperparameter tuning, as mentioned in the research highlights, further enhances its performance in accurately identifying instances of insider trading.

2.2 OPTUNA: Hyperparameter Optimization Framework

OPTUNA serves as a hyperparameter optimization framework designed to assist in discovering the most suitable hyperparameter configurations for machine learning models. These hyperparameters are parameters that are predetermined and not learned during the training phase, encompassing aspects like learning rate, batch size, or the number of hidden layers in a neural network. Choosing suitable values for these hyperparameters is crucial for achieving optimal model performance.

OPTUNA simplifies the process of hyperparameter optimization by automating the search for the best hyperparameter configuration. It uses techniques such as Bayesian optimization and Pruned Tree of Parzen Estimators (TPE) to intelligently explore the hyperparameter space and select the most promising configurations. By repeatedly evaluating different combinations of hyperparameters, OPTUNA identifies the set that maximizes the performance metric of interest, such as accuracy or loss. This automated approach saves significant time and

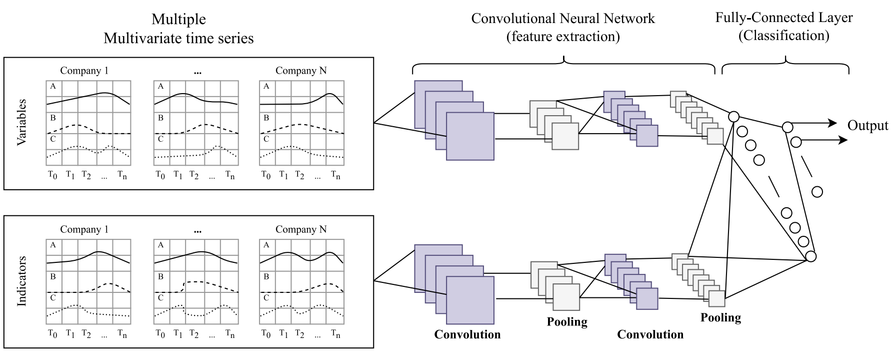

Fig. 1 Multi-channels CNN structure for multiple multivariate time series financial data classification

effort compared to manual tuning, allowing researchers and practitioners to quickly find optimal hyperparameter settings for their machine learning models.

In the context of detecting insider trading, OPTUNA, a hyperparameter optimization framework, is employed in conjunction with a multi-channel deep convolutional neural network (MTC-CNN). The multi-channel CNN is designed to process multiple indicators and variables associated with trading data simultaneously, capturing complex relationships and patterns. OPTUNA automates the process of searching for the optimal set of hyperparameters for the multi-channel CNN, such as learning rates, filter sizes, or the number of channels. By intelligently exploring the hyperparameter space and evaluating different configurations, OPTUNA identifies the best combination that maximizes the accuracy of insider trading detection. This integration of OPTUNA with the multi-channel CNN streamlines the hyperparameter tuning process, enhancing the performance and effectiveness of the expert system in identifying instances of insider trading.

3 Proposed Method

The insider trading samples for experiments are obtained from SEBI (Securities and Exchange Board of India, https://www.sebi.gov.in/), BSE (Bombay Stock Exchange, https://www.bseindia.com/), and NSE (National Stock Exchange of India Ltd., https://www.nseindia.com/), and the related indicators of insider trading samples are obtained from the CMIE (Centre for Monitoring Indian Economy, https://www.cmie.com/). Using the multi-channel CNN algorithm, the proposed method creates a model that can distinguish between samples containing insider trading and those without. Figure 2 depicts the model structure of the intelligent insider trading identification system. The system comprises five key components:

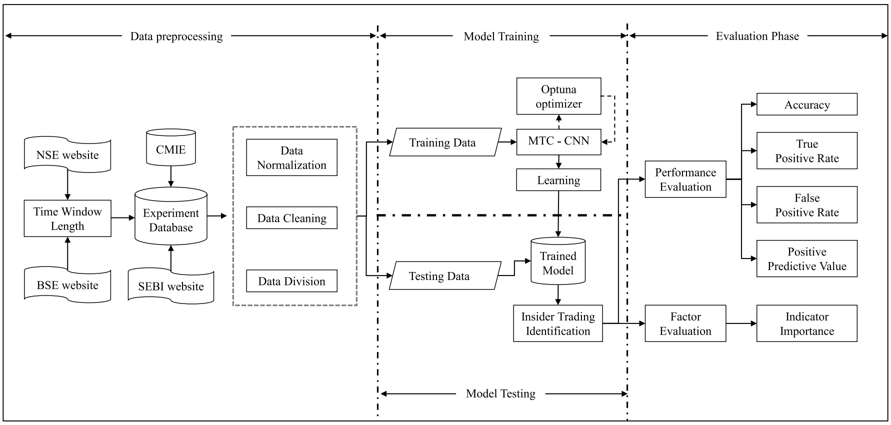

Fig. 2 Illustration of the proposed approach for identifying insider trading in the Indian stock market

- Pre-processing and Data Collection: We obtained information on insider trading samples from a number of sources, including the official websites of SEBI, NSE, BSE, and CMIE's Prowess IQ database. The data was collected for different time windows: 30, 60, and 90 trading days preceding the disclosure of insider information.
- Feature Selection: In this stage, a comprehensive set of indicators and variables associated with insider trading is considered. From this large pool of potential features, a selection process is performed to identify the most informative and relevant variables and indicators. Various techniques, such as statistical analysis and correlation analysis, are used to evaluate the significance of each feature in identifying insider trading. The aim is to narrow down the feature set to a manageable number of variables and indicators that contribute the most to the detection process.
- Model Training: The MTC-CNN model is trained using both insider trading and non-insider trading samples, incorporating indicators within different time window lengths of 30, 60, and 90 days. The Multi-channel CNN model is a crucial component of the system. It consists of multiple channels, with each channel representing a specific indicator or variable selected from the previous step. The Multi-channel CNN leverages the power of deep learning and convolutional operations to learn complex patterns and relationships within the input data. The architecture typically includes convolutional layers, pooling layers, and fully-connected layers, as discussed earlier. The network is optimized to identify and extract features related to insider trading from the multi-channel input.
- Model Optimization: This component involves the integration of OPTUNA, a hyperparameter optimization framework, into the system. OPTUNA automates the process of finding the optimal set of hyperparameters for the Multi-channel CNN. Hyperparameters, such as learning rates, filter sizes, and dropout rates, significantly impact the performance of the model. OPTUNA intelligently explores the hyperparameter space using techniques like Bayesian optimization or TPE and evaluates different configurations to find the combination that maximizes the accuracy of detecting insider trading. This automated tuning process enhances the performance of the expert system.
- Performance Evaluation: The final component focuses on using the optimized MTC-CNN to detect insider trading activities. The trained model takes preprocessed data as input and applies the learned features and patterns to identify instances of insider trading. The system produces classification results or predictions indicating whether a particular trading activity is likely to be insider trading or not. The effectiveness of the system in accurately identifying insider trading activities is evaluated using suitable evaluation metrics, including accuracy, precision, recall, and F1-score. These metrics help assess the system's performance in detecting insider trading.

This section outlines our step-by-step approach for pinpointing insider trading in the Indian stock market, as illustrated in Fig. 2. The figure acts as a visual guide, summarizing key stages discussed further in Sects. 4.2 (system setup and implementation) and 4.3 (benchmark method). From refining data to evaluating our model, the figure gives a quick peek into our methodology, emphasizing the crucial role of the multi-channel convolutional neural network (MTC-CNN) and the Optuna optimization framework.

4 Experiment Design

The experiment design outlines the methodology and procedures employed to evaluate the effectiveness of the proposed intelligent system in identifying insider trading. This section describes the experiment data used, the benchmark method employed for comparison, and the evaluation criteria utilized to assess the system's performance.

4.1 Experiment Data

The experiment data refers to the dataset used for training, testing, and validating the proposed intelligent system. The study employed a dataset encompassing insider trading cases spanning from 2001 to 2020, sourced from the SEBI website (https://www.sebi.gov.in/). To ensure balance, an equivalent number of non-insider samples were also incorporated, creating a 1:1 ratio of illegal insider trading to non-insider trading instances. In total, there were 30 cases, with 15 instances of insider trading and 15 instances of non-insider trading. The selection criteria for these cases were as follows: (1) the listed companies belonged to the same industry as the insider trading cases, and (2) the non-insider trading cases had no history of insider trading activity. This selection process ensures that the samples accurately represent the two categories. For variable channel, an individual case is defined with a matrix of 43×30 for 30-days, 43×60 for 60-days, and 43×90 for 90-days of window length and similar for the indicator channel.

Stock exchange websites BSE (https://www.bseindia.com/), NSE (https://www.nseindia.com/), and CMIE's (https://www.cmie.com/) Prowess IQ database were used to collect equity data for all samples (SEBI, https://www.sebi.gov.in/). Variables represent measurable quantities or characteristics, while indicators are derived metrics or signals that provide insights into specific aspects of insider trading. To make sure the data is reliable and suitable for analysis, it goes through a process called pre-processing that involves organizing and checking the data to ensure its quality. To test the multicollinearity between the variables, a Pearson's correlation coefficient (PCC) statistical test is carried out with a cutoff of 0.70 for identifying and discarding irrelevant features. To boost our model's efficiency and reliability, we applied the min–max normalization method for scaling features to account for a significant number of market fluctuations and trade noises. A total of 96 features (variables and indicators) were collected out of which 10 features were excluded from the analysis due to insufficient observations, multicollinearity, and the requirement to split these variables and indicators equally making it a total of 86 features. When working with a small dataset, it becomes challenging to split the data into traditional training, testing, and validation sets while ensuring reliable model evaluation. K-fold cross validation is known for maximizing data utilization, balance bias and variance in performance evaluation, and detecting overfitting or underfitting. This technique enables a thorough evaluation of our method's performance and generalizability, showcasing our commitment to extracting the most from our limited samples. Hence K-fold cross-validation technique has been used that repeatedly split the data into training and testing allowing a more comprehensive evaluation by leveraging all available data. The aim of this study is to effectively investigate and detect insider trading activities in the Indian market using this carefully curated dataset and suitable techniques (Table 1).

4.2 System Setup and Implementation

After performing dataset pre-processing, the hyperparameters of the CNN model were fine-tuned. The model was trained using optimized parameters. We have optimized the multi-channel convolutional neural network (MTC-CNN) using the Optuna optimization framework. The goal of this optimization process is to find the best set of hyperparameters that yield the highest performance for our MTC-CNN model. The optimization process, resulted in the selection of a convolutional layer with a (3,3) kernel size and ReLU activation function as part of the best hyperparameter set. Figure 3 shows the initial setting of parameters of the Optuna framework that we used for hyper-parameter tuning. The MTC-CNN model is designed to process time-series data with different time windows, namely 30 days, 60 days, and 90 days. The pseudo code for the proposed MTC-CNN model is shown in Algorithm 1.

Table 1 A list of features (variables and indicators) for insider trading

Price-related	Prev close; Open price; High price; Low price; Last traded price; Close price
Volume-related	Total traded quantity; No. of orders; Deliverable quantity; On-balance volume (OBV)
Trend and direction	Close location value (CLV); Accumulation/distribution (AD); Difference moving up (DM+); Difference moving down (DM−); True range (TR); Directional trend (DM14+); Directional trend (DM14−); Directional indicator (DI14+); Directional indicator (DI14−); Directional index (DX); Average directional movement index (ADX); Exponential moving average (EMA)
Volatility and bands	Sigma coefficient; Standard deviation (σ30); Bollinger upper band; Bollinger lower band; Chaikin money flow (CHMF); Chaikin volatility (CHV)
Oscillators	Commodity channel index (CCI); Full stochastic %K (Fast FS); Full stochastic %D (Full FS); Price oscillator (POS); Relative strength index (RSI); Relative strength (RS)
Risk-related	Security VaR; Nifty 50 Close price; Index value at risk (IVaR); Floating stock turnover rate (FSTR); Total share turnover rate (TSTR); Excess return compared with same market (ERCSM)
Miscellaneous	Typical price (TP); Detrended price oscillator (DPO); Ease of movement (EMV); Fibonacci retracement (FR); Fibonacci extension (FE); Money flow index (MFI); Moving average convergence divergence (MACD); Pivot point (PP); First support (S1); Second support (S2); Third support (S3); First resistance (R1); Second resistance (R2); Third resistance (R3); Rate of change (ROC); Williams %R (%R)

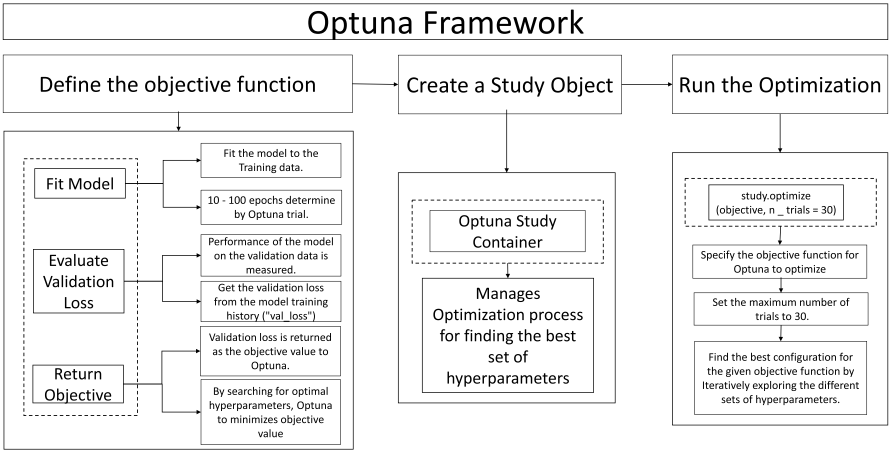

Fig. 3 Optuna framework and parameter optimization

Algorithm 1 Multichannel convolutional neural network (MTC-CNN)

algorithm Multichannel-CNN
input: input1: 3D matrix (30, 43, 1), input2: 3D matrix (30, 43, 1)
output: Score of Multichannel-CNN trained model on the test dataset
let conv1 be the result of $Conv2D(32, kernel\_size = (3,3), activation =' relu')$(input1)
let conv2 be the result of $Conv2D(32, kernel\_size = (3,3), activation =' relu')$(input2)
let concat be the result of $Concatenate()$([conv1, conv2])
let dropout1 be the result of $Dropout(0.5)$(concat)
let flat be the result of $Flatten()$(dropout1)
let fc be the result of $Dense(64, activation =' relu')$(flat)
let dropout2 be the result of $Dropout(0.5)$(fc)
let outdense be the result of $Dense(1, activation =' sigmoid')$(dropout2)
let model be the result of Model(inputs=[input1, input2], outputs=outdense)
Compile model using $optimizer =' adam', loss =' binary\_cross\_entropy', metrics = ['accuracy']$
let f be the featureset 3D matrix
for each sample i in the dataset **do**
let f_i be the featureset matrix of sample i
for each channel j in i **do**
let V_j be the result of $Conv2D(32, kernel\_size = (3,3), activation =' relu')(j)$
append V_j *to* f_i
append f_i *to* f
$f_{train}, f_{test}, I_{train}, I_{test}$ be the result of splitting feature set and labels into train subset and test subset
let M be the result of creating Multichannel-CNN model using (f_{train}, I_{train})
let score be the result of evaluating the model on test data (f_{test}, I_{test})
return score

Each time window configuration has different hyperparameters for its layers shown in the Table 2. For each time window configuration, the model consists of the following layers:

Convolutional Layer: This layer performs convolutional operations on the input data. The number of channels (filters) in this layer is different for each time window. The number of channels is 52 for the 30-day window, 22 for the 60-day window, and 19 for the 90-day window. The kernel size used for convolution is (3,3), which means a 3×3 filter is applied to the input.

Dropout_1: Dropout serves as a regularization method where, during training, a portion of input units are randomly assigned a value of 0. This process aids in mitigating overfitting. Specifically, a dropout rate of 0.5 is utilized, implying that every neuron in the preceding layer has a 50% probability of being excluded during training.

Fully Connected Layer: This layer is also known as the Dense layer and is responsible for connecting all neurons from the previous layer to the next layer. The number of neurons in this layer varies for each time window. There are 58 neurons for the 30-day window, 59 neurons for the 60-day window, and 25 neurons for the 90-day window. The activation function used for this layer is ReLU (Rectified Linear Unit), which introduces non-linearity into the model.

Dropout_2: This is another dropout layer, similar to Dropout_1, with a dropout rate of 0.5.

Final Output Layer: The final layer of the model consists of a single neuron that represents the output. To ensure its suitability for binary classification tasks, the Sigmoid function is used as the activation function for this layer. It effectively squashes the output value within the range of 0–1.

By optimizing these hyperparameters using the Optuna framework, we have found the best configuration for each time window that maximizes the performance of our multi-channel CNN model for the detection of insider trading in the Indian stock market.

4.3 Benchmark Method

The benchmark method serves as a baseline or reference for comparing the performance of the proposed intelligent system. In this study, the main objective is to

Table 2 MTC-CNN optimized parameter

Time window (days)	Convolution layer			Dropout_1	Fully connected layer		Dropout_2	Final output layer	
	Layer	Kernel size	Activation function		Dense	Activation function		Dense	Activation function
30	52	(3,3)	ReLU	0.5	58	ReLU	0.5	1	Sigmoid
60	22	(3,3)	ReLU	0.5	59	ReLU	0.5	1	Sigmoid
90	19	(3,3)	ReLU	0.5	25	ReLU	0.5	1	Sigmoid

detect insider trading using a deep learning model. The benchmark method includes both machine learning and deep learning approaches. For machine learning, random forest (RF) and logistic regression (LR) are commonly used as benchmark methods. As for deep learning, convolutional neural network (CNN) is considered as the benchmark method. The benchmark method is implemented using appropriate algorithms and techniques, and its performance is evaluated using the same dataset and evaluation criteria as the proposed intelligent system. The purpose is to assess how the proposed system compares to existing methods and whether it provides improvements in accuracy or other relevant metrics.

4.4 Evaluation Criteria

The evaluation criteria are the metrics used to assess the performance of the proposed intelligent system and the benchmark method. These metrics provide quantitative measures to evaluate the system's ability to accurately identify insider trading activities. Common evaluation criteria in machine learning and data analysis include accuracy, precision, recall, and F1-score. Accuracy evaluates the overall correctness of the system's predictions. Precision measures the ratio of correctly identified insider trading instances to all instances predicted as insider trading. Recall assesses the proportion of correctly identified insider trading instances out of all actual insider trading instances. F1-score offers a balanced evaluation of precision and recall. The calculation of these evaluation criteria involves the use of mathematical equations:

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$$

$$\text{F1} - \text{score} = 2 * (\text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall})$$

Here, TP represents true positive (correctly predicted insider trading), TN represents true negative (correctly predicted non-insider trading), FP represents false positive (incorrectly predicted insider trading), and FN represents false negative (incorrectly predicted non-insider trading). The evaluation criteria provide a quantitative basis for comparing the performance of the proposed intelligent system and the benchmark method, facilitating an objective assessment of their effectiveness in detecting insider trading activities accurately.

5 Results

The performance of various models was assessed in an experiment to detect insider trading. The models evaluated included logistic regression (LR), random forest (RF), convolutional neural network (CNN), and the proposed MTC-CNN. The

Table 3 Accuracy results

Time window (days)	Random forest (RF) (%)	Logistic regression (LR) (%)	Convolutional neural network (CNN) (%)	MTC-CNN-proposed (%)	MTC-CNN-proposed with OPTUNA (%)
30	71.42	57.14	75.00	87.50	87.50
60	57.14	57.14	62.50	75.00	75.00
90	71.42	71.42	50.00	62.50	75.00

Table 4 Precision results

Time window (days)	Random forest (RF) (%)	Logistic regression (LR) (%)	Convolutional neural network (CNN) (%)	MTC-CNN-proposed (%)	MTC-CNN-proposed with OPTUNA (%)
30	50.00	33.33	66.66	75.00	100.00
60	66.66	100.00	50.00	100.00	66.66
90	100.00	66.66	42.85	50.00	60.00

evaluation was carried out using three distinct time windows: 30 days, 60 days, and 90 days. The assessment of the models' performance employed several evaluation metrics, such as accuracy, precision, recall, F1-score, AUC ROC, and false positive rate (FPR).

5.1 Accuracy and Precision Result

The accuracy and precision results for detecting insider trading using different models across various time windows are presented in Tables 3 and 4.

For the 30-day time window, both the MTC-CNN-Proposed model and the MTC-CNN-Proposed model with Optuna optimization perform better than the RF, LR, and CNN models, shown in Table 3. They achieve higher accuracy of 87.50% and have lower rates of false positives compared to other models. The precision of the MTC-CNN-Proposed model is 75.00%, as shown in Table 4, indicating a good proportion of correctly predicted insider trading cases. However, the MTC-CNN-Proposed model with Optuna optimization achieves even higher precision of 100%, suggesting that it has an excellent ability to accurately identify instances of insider trading.

Moving to the 60-day time window, both the MTC-CNN-Proposed model and the MTC-CNN-Proposed model with Optuna optimization maintain their superior performance compared to the RF, LR, and CNN models, as shown in Tables 3 and 4. They achieve an accuracy of 75.00%, outperforming other models. The MTC-CNN-proposed model demonstrates perfect precision of 100.00%, indicating that all positive predictions of insider trading were correct. The MTC-CNN-proposed model with Optuna optimization achieves a precision of 66.66%, showing its ability to accurately classify insider trading cases. In contrast, the RF, LR, and CNN models achieve precision values of 66.66%, 100.00%, and 50.00% respectively, highlighting the better performance of the MTC-CNN-proposed model with Optuna optimization.

At the 90-day time window, the MTC-CNN-Proposed model achieves an accuracy of 62.50% and with Optuna optimization, it improves to 75%, as seen in Table 3. This outperforms the other models, as shown in Table 1. However, the precision of the MTC-CNN-Proposed model decreases to 50.00%, as seen in Table 4, indicating a higher rate of false positives compared to some benchmark models. Nevertheless, the MTC-CNN-Proposed model still performs better than the CNN model in terms of precision. Notably, when incorporating the Optuna framework, the MTC-CNN-Proposed model achieves a precision of 60.00%. These findings highlight the effectiveness of the MTC-CNN-proposed model, particularly when optimized with Optuna, in accurately classifying insider trading cases in the Indian market.

5.2 Recall, F1-Score, AUC ROC, and FPR result

The evaluation metrics of Recall, F1-score, AUC ROC, and FPR provide further insights into the performance of different models in detecting insider trading across different time windows.

At 30 days, as shown in the Table 5, the MTC-CNN-Proposed model achieves a high Recall of 100.00%, indicating its ability to correctly identify all actual insider trading cases. It also demonstrates a competitive F1-score of 85.71% and an AUC ROC of 90.00%, suggesting a good balance between precision and recall.

Table 5 Performance comparison of models for insider trading classification

Time window (days)	Model	Recall (%)	F1-score (%)	AUC ROC (%)	FPR (%)
30	RF	50.00	50.00	65.00	20.00
	LR	50.00	40.00	54.99	40.00
	CNN	66.66	66.66	73.33	20.00
	MTC-CNN-proposed	100.00	85.71	90.00	20.00
	MTC-CNN-proposed with optuna	66.66	80.00	83.33	0.00
60	RF	50.00	57.14	58.33	33.33
	LR	25.00	40.00	62.50	0.00
	CNN	100.00	66.66	70.00	60.00
	MTC-CNN-proposed	33.33	50.00	66.66	20.00
	MTC-CNN-proposed with optuna	66.66	66.66	73.33	0.00
90	RF	33.33	50.00	66.66	0.00
	LR	66.66	66.66	70.83	25.00
	CNN	100.00	60.00	60.00	20.00
	MTC-CNN-proposed	100.00	66.66	70.00	60.00
	MTC-CNN-proposed with optuna	100.00	75.00	80.00	40.00

Additionally, the model has a low FPR of 20.00%, indicating a low rate of incorrectly classifying non-insider trading cases as positive. The performance of the MTC-CNN-Proposed with Optuna Framework is also significant, with high values for Recall, F1-score, and AUC ROC, as well as an impressive FPR of 0.00%, indicating no false positives.

Moving to the 60-day time window, the MTC-CNN-Proposed model demonstrates a Recall of 33.33% and a F1-score of 50.00%, suggesting moderate performance in identifying insider trading cases. The AUC ROC value of 66.66% indicates a fair ability to distinguish between positive and negative samples. The model achieves a low FPR of 20.00%, indicating a relatively low rate of false positives. When incorporating the Optuna framework, the MTC-CNN-Proposed with Optuna Framework model shows improved performance in Table 5, particularly in terms of Recall (66.66%), F1-score (66.66%), and AUC ROC (73.33%).

At the 90-day time window, the MTC-CNN-Proposed and MTC-CNN-Proposed with Optuna Framework models consistently demonstrate high Recall values of 100.00%, indicating their ability to capture all insider trading cases. The F1-scores range from 60.00 to 75.00%, indicating a reasonable balance between precision and recall. The AUC ROC values for these models range from 70.00 to 80.00%, indicating good discriminatory power. The FPR values vary from 20.00 to 60.00%, with the MTC-CNN-Proposed with Optuna Framework model achieving the lowest FPR.

Overall, the Table 5 highlights the superior performance of the MTC-CNN-Proposed model and the further improvement achieved by incorporating the Optuna framework. These models consistently demonstrate high Recall, F1-score, and AUC ROC values, indicating their effectiveness in detecting insider trading activities with minimal false positives.

5.3 Time Window Lengths Comparison and Indicator Importance

Time window lengths and indicator importance play a crucial role in insider trading detection. Let us examine the results for different time window lengths (30 days, 60 days, and 90 days) and their impact on the models' performance.

For the 30-day time window, the MTC-CNN-proposed model, especially when optimized with the Optuna framework, consistently outperformed the benchmark models (RF, LR, and CNN) in terms of accuracy, precision, recall, F1-score, and AUC ROC. This suggests that the MTC-CNN-Proposed model is effective in identifying insider trading activities within a relatively short time frame.

Moving to the 60-day time window, the performance of the models varied. While the MTC-CNN-Proposed model continued to outperform RF and LR in terms of accuracy, precision, recall, and F1-score, it achieved similar results as the CNN model. This indicates that as the time window increased, the performance of the models in detecting insider trading became more challenging.

Finally, for the 90-day time window, the results showed that the MTC-CNN-Proposed model, along with the MTC-CNN-Proposed with Optuna framework, consistently achieved higher accuracy, precision, recall, F1-score, and AUC ROC compared to the RF, LR, and CNN models. These findings suggest that the

MTC-CNN-Proposed model performs well even for longer time windows, demonstrating its robustness in detecting insider trading activities over an extended period.

The importance of different features was assessed to determine their significance in detecting insider trading in the Indian market, as depicted in the Fig. 3. The analysis covered three-time windows: 30 days, 60 days, and 90 days. Certain features consistently exhibited importance across all time windows. For instance, the "Open Price" feature had high importance scores of 15.05%, 12.17%, and 7.20% for the respective time windows. Another significant factor was the "average directional movement index (ADX)" with scores of 12.93%, 10.51%, and 9.03%. Similarly, the "moving average convergence divergence (MACD)" demonstrated consistent importance with scores of 9.38%, 8.16%, and 12.12%. The "relative strength (RS)" feature also proved to be relevant with scores of 4.39%, 9.87%, and 17.45%. In addition to these common features, others such as "Standard deviation (σ 30)", "Security VaR", "No. of Orders", and "Simple Moving Average (SMA)" played a role in detecting insider trading, albeit with varying importance across time windows. To enhance the analysis, features with less than 2% importance were filtered out. By considering the collective contribution of these important features, deep learning models can effectively identify suspicious trading patterns and enhance the detection of insider trading in the Indian market. These findings offer valuable insights into the crucial factors for detecting insider trading, thereby fostering market integrity and bolstering investor confidence (Figs. 4, 5, and 6).

Overall, the results indicate that the MTC-CNN-Proposed model, particularly when optimized with the Optuna framework, consistently outperforms the benchmark models across different time window lengths. This suggests its effectiveness in detecting insider trading activities, even as the time window increases. The choice of relevant indicators plays a crucial role in achieving accurate results and should be carefully considered in the model development process.

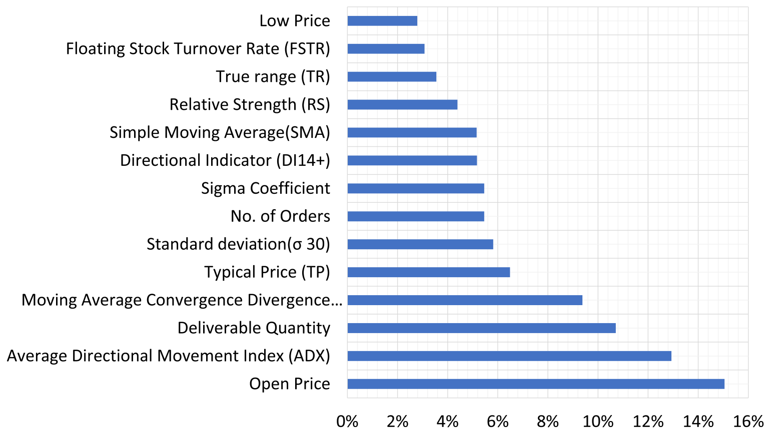

Fig. 4 Feature importance in detecting insider trading in the Indian market for 30 days

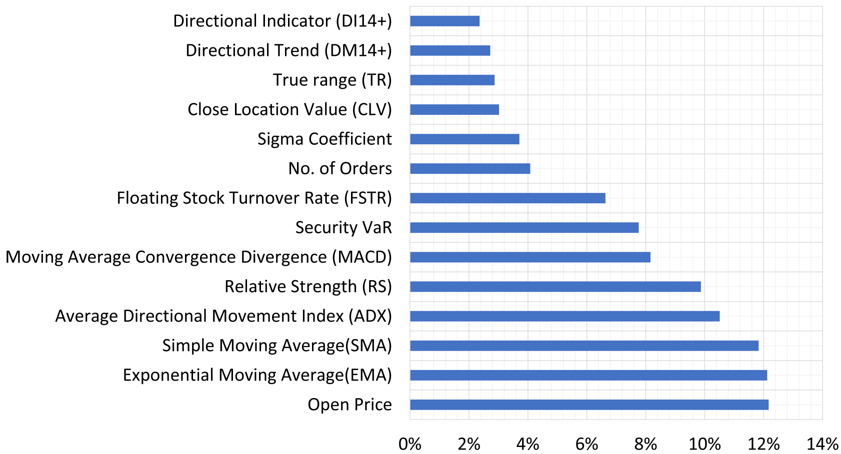

Fig. 5 Feature importance in detecting insider trading in the Indian market for 60 days

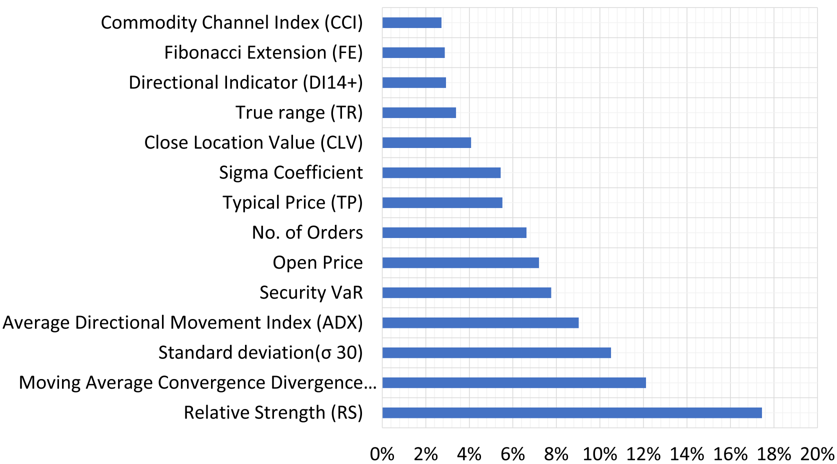

Fig. 6 Feature importance in detecting insider trading in the Indian market for 90 days

6 Conclusion

In conclusion, this study presents a comprehensive analysis of the impact of insider trading on unusual trading volume and information-driven returns in the Indian stock market. To effectively detect insider trading activities, we propose an intelligent deep learning model called the optimized MTC-CNN, which is augmented with Optuna hyperparameter optimization. The model showcases superior performance compared to benchmark models, such as RF, LR, and CNN, across various time windows (30, 60, and 90 days).

To ensure the reliability of our findings, we meticulously curated a dataset from reputable government sources, comprising 30 samples with equal representations of both insider trading and non-insider trading cases. The dataset encompasses 86 features derived from 43 trading-related variables and 43 indicators, all subjected to rigorous pre-processing to maintain data quality.

Our proposed model exhibits remarkable accuracy rates of 87.50% at 30 days, 75.00% at 60 days, and 62.50% at 90 days, signifying its potential as a promising tool for identifying insider trading activities in the Indian stock market. The model's ability to analyze data across different time windows allows for real-time monitoring of trading activities, contributing significantly to maintaining trust and confidence in the market.

The implications of our findings extend to market regulators, as our intelligent system aids in identifying fraudulent activities and safeguarding investor interests. By addressing the pressing issue of insider trading, our research contributes to the integrity of the market and assists in preserving a fair-trading environment.

However, it is essential to acknowledge the limitations of our study. We solely focused on publicly available investigation cases published by SEBI over a 20-year period (2001–2020) to examine insider trading. Future research should consider incorporating data from additional sources and expanding the time frame to further validate the model's robustness and generalizability.

Funding The author(s) received no financial support for the research, authorship, and/or publication of this article.

Data Availability The author confirms that all data generated or analysed during this study are available in the public domain website of SEBI (https://www.sebi.gov.in/), BSE (https://www.bseindia.com/), NSE (https://www.nseindia.com/), and CMIE (https://www.cmie.com/).

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this manuscript.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019) Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, in KDD '19* (pp. 2623–2631). New York: Association for Computing Machinery. https://doi.org/10.1145/3292500.3330701

Arifin, M. M., Based, M. A., Mumenin, K. M., Imran, A., Azim, M. A., Alom, Z., & Awal, M. A. (2021). OLGBM: Optuna optimized light gradient boosting machine for intrusion detection. In *2021 International conference on computer, communication, chemical, materials and electronic engineering (IC4ME2)* (pp. 1–4). https://doi.org/10.1109/IC4ME253898.2021.9768555

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE*. https://doi.org/10.1371/journal.pone.0180944

BSE—Bombay Stock Exchange of India. [Online]. Available: https://www.bseindia.com/. Accessed 04 July 2023

Cai, X., Hu, S., & Lin, X. (2012). Feature extraction using restricted Boltzmann machine for stock price prediction. In *2012 IEEE international conference on computer science and automation engineering (CSAE)* (pp. 80–83). https://doi.org/10.1109/CSAE.2012.6272913

Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE international conference on big data (big data)* (pp. 2823–2824). https://doi.org/10.1109/BigData.2015.7364089

CMIE—Centre for Monitoring Indian Economy Pvt. Ltd. [Online]. Available: https://www.cmie.com/. Accessed 04 July 2023

Deng, S., Wang, C., Fu, Z., & Wang, M. (2021). An intelligent system for insider trading identification in Chinese security market. *Computational Economics, 57*(2), 593–616. https://doi.org/10.1007/s10614-020-09970-8

Deng, S., Wang, C., Wang, M., & Sun, Z. (2019). A gradient boosting decision tree approach for insider trading identification: An empirical model evaluation of China stock market. *Applied Soft Computing, 83*, 105652. https://doi.org/10.1016/j.asoc.2019.105652

Di Persio, L., & Honchar, O. (2016) *Artificial neural networks architectures for stock price prediction: Comparisons and applications*. [Online]. Available: https://iris.univr.it/handle/11562/955101. Accessed 23 July 2023

Esen, M. F. (2020). A robust multivariate outlier detection method for detection of securities fraud. *IJBAN, 7*(3), 12–29. https://doi.org/10.4018/IJBAN.2020070102

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Gunduz, H., Yaslan, Y., & Cataltepe, Z. (2017). Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. *Knowledge-Based Systems, 137*, 138–148. https://doi.org/10.1016/j.knosys.2017.09.023

Hamester, D., Barros, P., & Wermter, S. (2015). Face expression recognition with a 2-channel convolutional neural network. In *2015 International joint conference on neural networks (IJCNN)* (pp. 1–8). https://doi.org/10.1109/IJCNN.2015.7280539

Hiransha, M. E., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. *Procedia Computer Science, 132*, 1351–1362. https://doi.org/10.1016/j.procs.2018.05.050

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications, 129*, 273–285. https://doi.org/10.1016/j.eswa.2019.03.029

Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. *Applied System Innovation, 4*(1), 9. https://doi.org/10.3390/asi4010009

Islam, S. R. (2018). A deep learning based illegal insider-trading detection and prediction technique in stock market. *Undefined*. [Online]. Available: https://www.semanticscholar.org/paper/A-Deep-Learning-Based-Illegal-Insider-Trading-and-Islam/ffb4bf38805fdf58bcd3aba7829b379996f24059. Accessed 31 January 2022

Jiang, J.-R., Lee, J.-E., & Zeng, Y.-M. (2020). Time series multiple channel convolutional neural network with attention-based long short-term memory for predicting bearing remaining useful life. *Sensors, 20*(1), 166. https://doi.org/10.3390/s20010166

Kumar, A., Singh, J. P., Rana, N. P., & Dwivedi, Y. K. (2023). Multi-channel convolutional neural network for the identification of eyewitness tweets of disaster. *Information Systems Frontiers, 25*(4), 1589–1604. https://doi.org/10.1007/s10796-022-10309-x

Liu, T., Zheng, H., & Liang, W. (2015). Parallel training of convolutional neural networks for small sample learning. In *2015 International joint conference on neural networks (IJCNN)* (pp. 1–6). Killarney: IEEE. https://doi.org/10.1109/IJCNN.2015.7280367

Messner, E., Fediuk, M., Swatek, P., Scheidl, S., Smolle-Jüttner, F.-M., Olschewski, H., & Pernkopf, F. (2020). Multi-channel lung sound classification with convolutional recurrent neural networks. *Computers in Biology and Medicine, 122*, 103831. https://doi.org/10.1016/j.compbiomed.2020.103831

NSE—National Stock Exchange of India Ltd. [Online]. Available: https://www.nseindia.com/. Accessed 04 July 2023

Park, Y. S., & Lee, J. (2010). Detecting insider trading: The theory and validation in Korea exchange. *Journal of Banking and Finance, 34*(9), 2110–2120. https://doi.org/10.1016/j.jbankfin.2010.01.018

Securities and Exchange Board of India. [Online]. Available: https://www.sebi.gov.in/. Accessed 09 April 2023

Seth, T., & Chaudhary, V. (2020). A predictive analytics framework for insider trading events. In *2020 IEEE international conference on big data (big data)* (pp. 218–225). https://doi.org/10.1109/BigData50022.2020.9377791

Shajihan, S. A. V., Wang, S., Zhai, G., & Spencer, B. F., Jr. (2022). CNN based data anomaly detection using multi-channel imagery for structural health monitoring. *Smart Structures and Systems, 29*(1), 181–193.

Shekhar, S., Bansode, A., & Salim, A. (2021). A comparative study of hyper-parameter optimization tools. In *2021 IEEE Asia-Pacific conference on computer science and data engineering (CSDE)* (pp. 1–6). https://doi.org/10.1109/CSDE53843.2021.9718485

Srinivas, P., & Katarya, R. (2022). hyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost. *Biomedical Signal Processing and Control, 73*, 103456. https://doi.org/10.1016/j.bspc.2021.103456

Yong, B. X., Abdul Rahim, M. R., & Abdullah, A. S. (2017). A stock market trading system using deep neural network. Communications in computer and information scienceIn M. S. Mohamed Ali, H. Wahid, N. A. Mohd Subha, S. Sahlan, M. A. M. Yunus, & A. R. Wahap (Eds.), *Modeling, design and simulation of systems* (pp. 356–364). Springer. https://doi.org/10.1007/978-981-10-6463-0_31

Zhang, X., Hao, Y., Shangguan, H., Zhang, P., & Wang, A. (2020). Detection of surface defects on solar cells by fusing multi-channel convolution neural networks. *Infrared Physics and Technology, 108*, 103334. https://doi.org/10.1016/j.infrared.2020.103334

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.